Tsinghua Science and Technology
Doi: 10.26599/TST.2026.90100019

Research Article

O^2Exp: Online Object Exploration in Underwater Environment

Xingyu Chen*, Yue Lu*, Shaoan Wang, Zhengxing Wu, and Junzhi Yu (✉)

Abstract The underwater environment contains a wealth of biological and mineral resources, making the deployment of autonomous underwater vehicles (AUVs) essential for exploration and development. Despite years of research in data-driven machine vision techniques, the offline collection of underwater data remains quite difficult compared to terrestrial samples. This paper focuses on online object exploration in underwater environments without manual intervention, including sub-tasks of close- and open-set detection, fine-grained novel-class subdivision, and few-shot incremental learning. To address this challenge, we start with a few-shot detector for detecting known classes and propose an open-set detector for exploring novel categories. The open-set detector can model unseen objects with fused semantics-localization cues and discrepancy-enhanced representation. Furthermore, we design detector-driven clustering to subdivide novel objects into an arbitrary number of novel classes as pseudo-labels. Finally, incremental learning is performed to model novel-category representation while maintaining base-class knowledge, where gradient rescaling and knowledge distillation strategies are designed to avoid catastrophic forgetting. Overall, our proposed framework, called O^2Exp, can autonomously explore objects in unstructured underwater environments. Extensive experiments with public datasets and real-world tests verify the accuracy, robustness, and practicality of the proposed O^2Exp framework.

Keywords autonomous underwater vehicle; object detection; machine vision; deep learning

• Xingyu Chen and Yue Lu equally contribute to this work.
• Xingyu Chen, Shaoan Wang, and Junzhi Yu are with the State Key Laboratory for Turbulence and Complex Systems, School of Advanced Manufacturing and Robotics, Peking University, Beijing 100871, China. Email: wangshaoan@stu.pku.edu.cn; yujunzhi@pku.edu.cn. Xingyu Chen is also with Zhongguancun Academy, Beijing 100094, China. Email: chenxingyu@bjzgca.edu.cn.
• Yue Lu and Zhengxing Wu are with the Laboratory of Cognitive and Decision Intelligence for Complex System, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. Email: yrqs1234@foxmail.com; zhengxing.wu@ia.ac.cn
Manuscript received: 2025-04-28; revised: 2025-12-30; accepted: 2026-01-16

1 Introduction

The underwater environments contain rich mineral and biological resources, but currently, only a small portion of them has been explored. In recent years, numerous studies have focused on autonomous underwater exploration and scene understanding using autonomous underwater vehicles (AUVs). Under this background, visual perception methods play an important role, and object detection is one of the crucial techniques to achieve intelligent understanding and analysis [1–3].

Existing object detection models are generally constrained to the predefined categories present in the training datasets. [4, 5]. For example, in autonomous grasping tasks, pre-defined classes of marine organisms are detected and grasped in the underwater environment [6]. However, compared to terrestrial scenes, acquiring image data in underwater environments is more challenging. Thus, available underwater detection datasets are limited and contain only a few object categories [7]. Therefore, existing object detection methods make it hard to accurately detect various novel objects from categories that were not included in the training set (i.e., unseen classes). As a result, autonomous visual exploration for open-world underwater scenes has rarely been studied.

To relieve the above-mentioned issue of data collection, many studies focus on open-set object detection. Dhamija et al., for the first time, formalize the problem of the open-set detection to detect objects beyond annotations [8]. In addition, Joseph et al. introduce a new challenge where the object detector is capable of incremental learning [9], combining open-set and incremental object detection as an open-world detection task. However, in previous studies, the training samples of incremental learning are offline labeled with manual efforts, hindering the formation of an autonomous system [9]. Therefore, a purely online task for object detection should be investigated for real-world autonomous exploration.

In this paper, for accurate detection of both known and novel objects during autonomous underwater perception, we study the task of online object exploration with four-fold objectives, as illustrated in Figure 1. First, detect known categories that are included in the training data. Second, detect novel categories beyond the training set, i.e., open-set

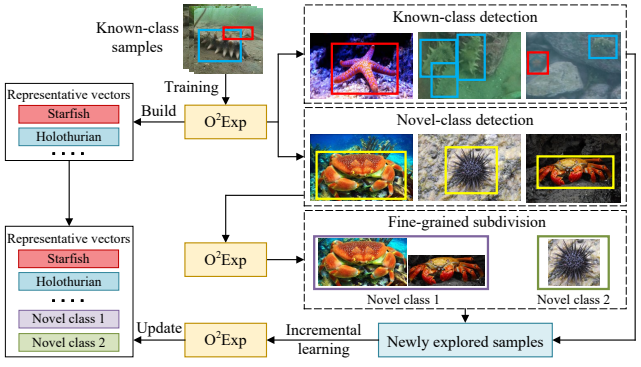

Fig. 1 Illustration of online object exploration task. The O^{2}Exp framework is designed for effective online object exploration with open-set detection, fine-grained novel-category subdivision, and few-shot incremental learning.

detection. Third, subdivide novel objects into fine-grained novel categories. Last, model novel categories with few-shot incremental learning. The whole process is fully autonomous without the need for manual intervention.

We develop a visual perception framework for the **O**nline **O**bject **Expl**oration task, namely O^{2}Exp, with close- and open-set detectors. We employ our previous work, Binary Similarity Few-Shot Detector (BSDet) [10], for the close-set detection task. In the meantime, BSDet is also responsible for constructing a representation library and performing incremental learning on newly discovered samples. To explore novel objects, we propose an open-set detector with semantics and localization cues (OpenSLD), where the novel-class detection performance is further improved by pseudo-label selection (PLS) and discrepancy enhancement module (DEM). Conventional open-set detectors can only classify novel-class objects as a single category of "unknown", without the ability to further differentiate them. To address this issue, we propose a detector-driven clustering (DetClust) that can produce an arbitrary number of novel categories. Besides, DetClust can be well integrated with BSDet-based object representation by using cosine similarity as the clustering metric. Furthermore, gradient rescaling and knowledge distillation strategies are designed to maintain base representations during few-shot incremental learning, avoiding catastrophic forgetting. Finally, we verify our framework with extensive experiments based on public datasets and real-world applications.

Our main contributions are summarized as follows:

- We propose an O^{2}Exp framework to solve the challenging problem of online object exploration, including known- and novel-class detection, fine-grained novel-category subdivision, and few-shot incremental learning.
- We propose OpenSLD for open-set object detection with both semantics and localization cues, where PLS and DEM are designed to improve the novel-class detection performance.
- We introduce a DetClust method to subdivide novel objects into fine-grained novel categories. We also develop gradient rescaling and distillation strategies to avoid knowledge forgetting during incremental learning of novel samples.
- The superiority of the proposed O^{2}Exp framework is validated by comparing related studies on public datasets and performing the task of online object exploration in real-world unstructured underwater environments.

Beyond existing offline algorithms (e.g., [11–13]), our framework achieves a systematic breakthrough by establishing a fully autonomous, real-time, online exploration paradigm for unstructured underwater environments, and a practical breakthrough by enabling AUVs to continuously evolve their knowledge base in the wild.

2 Related Work

2.1 Underwater Object Detection

Autonomous underwater vehicles with optical vision systems are commonly used to explore underwater environments and objects [14]. Despite advancements in deep learning, underwater object detection faces challenges such as low image quality, limited datasets, and domain shifts. Recent efforts have aimed to address these issues. Chen et al. analyzed detection mechanisms in optically degraded underwater environments, demonstrating the importance of image restoration for effective object detection, and an anchor-offset detection model for underwater objects was later proposed [15, 16]. Fan et al. enhanced detection performance by restoring degraded underwater images at the feature level [17]. Hua et al. proposed feature enhancement and progressive dynamic aggregation strategies to improve detection performance in underwater images [18]. The above methods follow the traditional detection paradigm, which is constrained by the data collection and annotation.

Rather than apply traditional detection methods within the underwater scenes, this paper aims to build a unified framework of online object exploration, including known- and novel-class detection, fine-grained novel-category subdivision, and incremental exploration for underwater objects.

2.2 Open-Set Object Detection

To address the challenge of detection beyond dataset, an open-set detection task is explored, enabling the detector to be aware of novel-class objects [8, 9, 19].

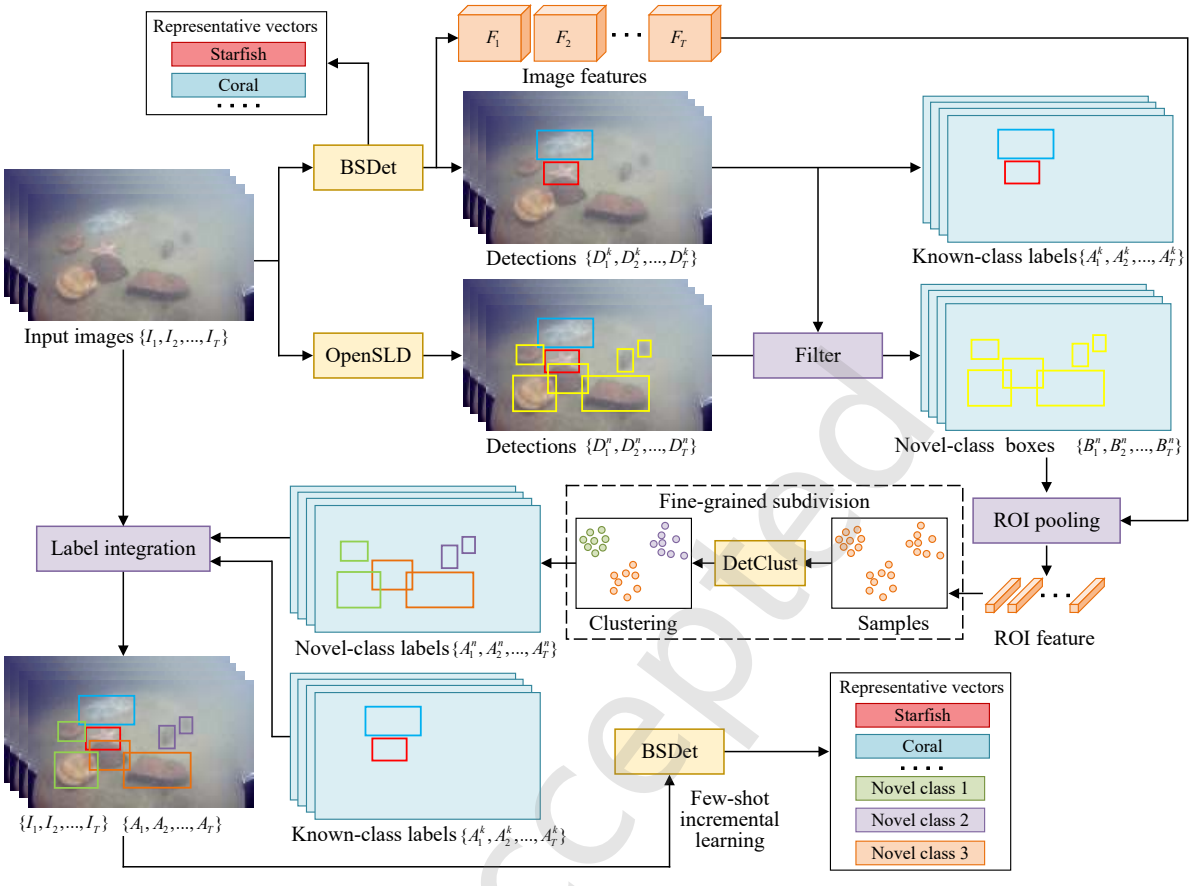

Fig. 2 Overview of our proposed O²Exp framework for online object exploration task. We employ BSDet and OpenSLD to detect known and novel objects, respectively. Then, a DetClust method is designed to subdivide newly discovered objects into fine-grained novel classes, which serve as pseudo labels for few-shot incremental learning.

Dhamija et al. formalize the problem and introduce evaluation metrics for this task [8]. Joseph et al. employed clustering- and energy-based approaches to distinguish known- and novel-class objects [19]. However, this method requires full annotation information during inference, which is infeasible for real-world applications. Based on DETR [20], Gupta et al. predicted novel classes based on the query embedding and introduced the OW-DETR detector for open-set detection [11]. OW-DETR introduces a foreground prediction branch to enhance the detection performance of novel classes by discriminating them from the background. However, OW-DETR relies entirely on the encoding features for pseudo-label selection without supervision and suffers from confusion issues between known- and novel-class objects. Kim et al. proposed an innovative region proposal approach, termed Object Localization Network (OLN) [21]. Diverging from RPN, OLN employs the centerness as a foreground confidence score. Nevertheless, owing to the absence of semantic guidance, OLN may inadvertently yield false negatives. Wu et al. proposed UC-OWOD that can detect objects in fine-grained novel classes [22], but the number of novel categories should be manually defined, making it unsuitable for online tasks.

In contrast, in this paper, we pose an online object exploration challenge and develop a visual perception framework for open-world object detection and exploration in a purely online manner. For open-set detection, we propose a semantics-localization fused region proposal and a discrepancy enhancement module to improve the awareness of novel-class objects.

2.3 Clustering

Clustering is an unsupervised algorithm aimed at partitioning samples into multiple clusters based on data similarity. Commonly used clustering methods include partition-based, hierarchical, density-based, and grid-based approaches. Partition-based methods aim to subdivide all samples into multiple clusters based on minimizing distances, where widely utilized algorithms include K-means [12], K-medoids [23], and PAM [24]. Hierarchical methods construct samples into a tree structure and then decompose them according to hierar-

chy through aggregation or division [25, 26]. Density-based methods utilize the density of sample points for clustering [27, 28]. Specifically, they determine whether the density of a sample point exceeds a density threshold and assign samples to the nearest cluster. Grid-based methods map samples into grid cells and determine their density within these cells [29]. Then, adjacent dense grid cells are grouped together. Recently, deep learning-based methods are studied by employing deep neural networks to extract high-level features for clustering [30, 31]. Because deep features have semantically expressive information, the clustering performance is improved.

In this paper, we use the idea of clustering for fine-grained novel-class subdivision.

2.4 Incremental learning

Incremental learning refers to the ability to continuously acquire new knowledge (i.e., novel categories) while retaining previously learned information. Popular methods can be divided into three types based on model structure, replay, and regularization. Model structure-based methods typically modify the model structure, such as expanding model parameters based on a topological graph structure [32, 33]. This allows the model parameters to have a larger capacity, thus maintaining recognition accuracy for known-class objects. However, modifying the model can increase computational and storage costs.

Regularization-based methods use knowledge distillation (KD) as a regularization term to prevent the model from forgetting [34–36]. Learning without Forgetting [34] first introduces distillation into incremental learning by constraining the new model to match the output logits of the old model, enabling knowledge preservation without storing old data. Subsequent methods enhance distillation by addressing class imbalance and representation drift. Hou et al. [37] perform feature-level distillation and adopt cosine-normalized classifiers to reduce bias toward new classes. Wu et al. [38] explicitly corrects classifier bias via a post-hoc calibration layer. More recent approaches, such as PODNet [35], extend distillation to multi-layer feature statistics, demonstrating that preserving intermediate representations is critical for long-term incremental performance. Overall, KD-based methods form the dominant paradigm for mitigating catastrophic forgetting in incremental learning. However, they require the inference of the base model during the process of learning novel-class objects, leading to increased training time.

Replay-based methods store samples or features of known-class objects and then train them together with novel-class samples to avoid knowledge forgetting [13, 39, 40]. Nonetheless, the online-discovered class usually has few-shot samples, and the replay mechanism is not effective enough for few-shot incremental learning.

In this paper, we introduce gradient scaling and knowledge distillation strategies to replay-based methods to preserve base knowledge under a few-shot condition.

3 Approach

For underwater object exploration, this paper proposes the O^{2}Exp framework, as illustrated in Figure 2. The framework consists of two detectors: BSDet [10] for detecting known-class objects (Section 3.1) and OpenSLD for detecting novel-class objects (Section 3.2). Besides, fine-grained novel-class clustering (Section 3.3) and few-shot incremental learning (Section 3.4) will be introduced for the modeling of novelly discovered categories. We redirect the readers to the Appendix for the workflow of O^{2}Exp.

3.1 Preliminary: Close-set Detection with BSDet

For few-shot object detection, a binary similarity detector (i.e., BSDet) has been previously designed. Specifically, BSDet models semantic categories with representative vectors $\mathbf{r}$, and the detection is based on the similarity between $\mathbf{r}$ and the RoI features. In addition, a binary similarity head is leveraged to pose the classification task as multiple binary similarity measurements rather than a multi-class prediction. Moreover, focusing on the hard negative samples, BSDet includes a feature enhancement module, which can push the features of positive and hard negative samples far away from each other during training and thus effectively suppress false positives. As a result, BSDet achieves superior performance on the few-shot object detection task. We draw the readers' attention to [10] for more details.

Built upon previous work BSDet [10], this paper transcends the role of a static detector by forming a closed-loop system that includes unknown discovery, fine-grained subdivision, and online knowledge evolution.

3.2 Open-Set Detection with OpenSLD

As shown in Figure 3, we propose an open-set object detection method based on Faster RCNN [41], called OpenSLD. For a given input image, the backbone network and FPN [42] perform multi-scale feature extraction. Also, we design fused RPN (FRPN) to predict region proposals and pseudo-label selection (PLS) to determine pseudo annotations for novel-class objects. After obtaining the proposal regions, RoI pooling is performed to extract RoI features, which are

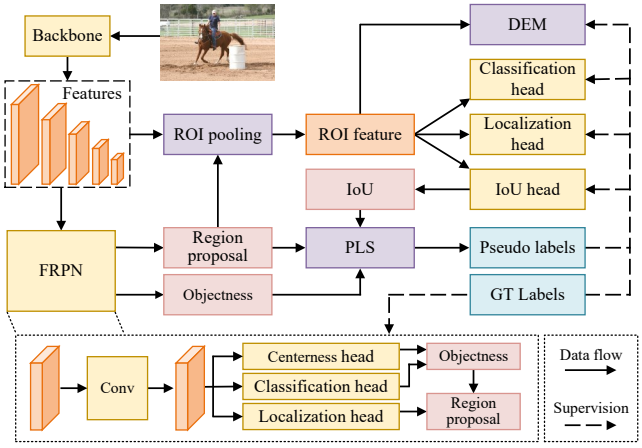

Fig. 3 The training process of OpenSLD.

then used for classification and localization predictions. The classification and localization heads are inherited from Faster RCNN. Additionally, we design a discrepancy enhancement module (DEM) to increase the contrast between known- or novel-class representations during training.

FRPN. RPN predicts foreground probability for each anchor box with only a classification cue, and it could assign potential novel-class objects as background during training, resulting in poor recall rates for novel-class objects during inference. In contrast, FRPN aims to predict centerness, classification, and localization for each anchor box, where the centerness is to represent the deviation between the predicted position and the object center.

By fusing the centerness $cent$ and classification confidence p_{fg}, we use objectness obj to represent the foreground confidence of a region proposal as follows,

$$obj = \sqrt{cent \cdot p_{fg}}. \tag{1}$$

During training, we utilize the binary cross-entropy loss function as the training objective:

$$L^{cls}_{FRPN} = -(1-p^*_{fg})\log(1-p_{fg}) - p^*_{fg}\log(p_{fg}), \tag{2}$$

where * denotes the ground truth. In addition, we design the ground truth of centerness as:

$$cent^* = \sqrt{\frac{\min(l^*, r^*)}{\max(l^*, r^*)} \times \frac{\min(t^*, b^*)}{\max(t^*, b^*)}}, \tag{3}$$

where l^*, r^*, t^*, b^* are left-, right-, top-, and bottom-distances between the anchor center and ground-truth bounding box. The training of both centerness and localization can be formulated with L1 loss:

$$\begin{aligned} L^{cent}_{FRPN} &= ||cent - cent^*||_1 \\ L^{loc}_{FRPN} &= ||\mathbf{b} - \mathbf{b}^*||_1 \end{aligned}, \tag{4}$$

Algorithm 1 Pseudo-label selection

[1] Set thresholds $\alpha_{IoU}, \alpha_{obj}, \alpha_{fuse}$. Apply FRPN to obtain region proposals R and obj. $\forall r \in R$, calculate IoU^{GT}_r. $R = \{r | r \in R, IoU^{GT}_r < \alpha_{IoU}, obj_r > \alpha_{obj}\}$ $\forall r \in R$, predict IoU value IoU_r. $conf_r = \sqrt{obj_r \cdot IoU_r}$. $R = \{r | r \in R, conf_r > \alpha_{fuse}\}$. Apply NMS to R. Apply localization head to R as the pseudo labels.

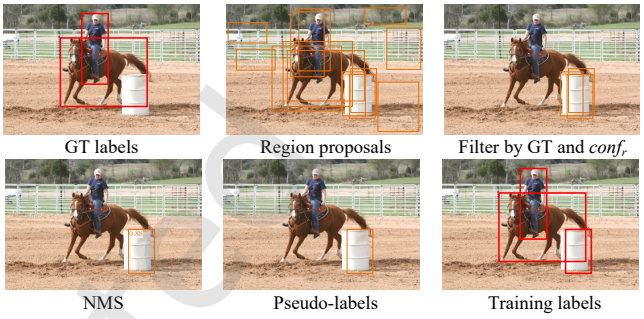

Fig. 4 The effect of PLS to construct training labels.

where $\mathbf{b}$ is the predicted bounding boxes. Finally, the loss function of FRPN can be given as:

$$L_{FRPN} = L^{cls}_{FRPN} + L^{loc}_{FRPN} + L^{cent}_{FRPN}. \tag{5}$$

PLS. For learning newly discovered objects, we construct pseudo-labels as their annotation. The workflow of PLS is presented in Algorithm 1. First, use FRPN to obtain region proposals R. Next, to ensure that generated pseudo-labels only contain potential novel-class objects, the pseudo-labels should have a small overlap with annotated boxes of known-class objects. Therefore, the candidates are filtered by pre-defined thresholds α_{IoU} and α_{obj}. To further improve the quality of region proposals, we use a fully connected layer to predict the IoU value for each candidate, then fuse IoU and objectness as the final confidence. Based on the confidence, threshold α_{fuse}, and NMS, the candidates are further filtered. Finally, we use the localization head to refine the reserved region proposals as the pseudo-labels. The effect of PLS is illustrated in Figure 4. The IoU branch is essential in PLS, and we use L1 loss for training, i.e., $L_{IoU} = ||IoU - IoU^*||_1$.

DEM. To alleviate the issue of confusion between known and novel classes, we propose DEM to provide constraints on similarity, energy, and classification in training.

First, we design similarity-based enhancement. In general, objects in the same category should have a higher level of feature similarity than those in different categories. However, the traditional classification task fails to model the similarity. Inspired by BSDet, we build a library of learnable represen-

tative vectors for known classes. For an ROI feature $\mathbf{x}$, we calculate the Gaussian-activated cosine similarity:

$$cos_i = \frac{\mathbf{x} \cdot \mathbf{r}_i}{||\mathbf{x}|| \cdot ||\mathbf{r}_i||}, \quad sim_i = \exp\left(-\eta(1-cos_i)^2\right), \quad (6)$$

where $\mathbf{r}_i$ is the representative vector for class i and η is a hyperparameter.

We use focal loss for training as follows,

$$L_{DE}^{sim\_cls} = \sum_{i=1}^{N_k+1} \begin{cases} -\beta(1-sim_i)^\gamma \log(sim_i), & l_i = 1 \\ -(1-\beta) sim_i^\gamma \log(1-sim_i), & l_i = 0 \end{cases}, \quad (7)$$

where β, γ are hyper-parameters, and l is the one-hot class vector where $l_i = 1$ at the index of ground-truth class i^*. Furthermore, we use the triplet loss to enhance inter-class similarity distance:

$$L_{DE}^{sim\_dist} = \text{ReLU}\left(\max_{i \neq i^*} cos_i - cos_{i^*} + \delta\right), \quad (8)$$

where δ represents the lower limit of the expected difference between different categories. The triplet loss can ensure the validity of representative vectors.

Second, we design an energy-based enhancement. In the popular energy model [43], an energy term $E(a, b)$ can measure the matching level between the inputs. Accordingly, we use the Gibbs distribution to obtain the classification probability for an ROI feature $\mathbf{x}$ and class i as follows,

$$p(i|\mathbf{x}) = \frac{e^{-E(\mathbf{x},i)}}{\int_j e^{(-E(\mathbf{x},j)}} = \frac{e^{-E(\mathbf{x},i)}}{e^{-E(\mathbf{x})}}. \quad (9)$$

Additionally, the classification head of OpenSLD can be denoted as $f(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^{N_f}$, where D is the dimension of $\mathbf{x}$, and $N_f = N_K + 2$ with N_K known classes, a novel class, and a background class. Based on the softmax function, the category probability predicted by the classification head is given as:

$$p(i|\mathbf{x}) = \frac{e^{f_i(\mathbf{x})}}{\sum_{j=1}^{N_f} e^{f_j(\mathbf{x})}}, \quad (10)$$

where $f_i(\mathbf{x})$ is the ith channel of $f(\mathbf{x})$. By comparing Eqs. (9) and (10), we set $E(x, i) = -f_i(x)$.

According to Helmholtz free energy [44], the energy of RoI features is given as:

$$\begin{aligned} E^k(\mathbf{x}) &= -\log \sum_{i=1}^{N_k} e^{-E(\mathbf{x},i)} \\ E^u(\mathbf{x}) &= -\log e^{-E(\mathbf{x},i)}|_{i=N_k+1} \end{aligned}, \quad (11)$$

where $E^k(\mathbf{x})$ and $E^u(\mathbf{x})$ denote the level of match between $\mathbf{x}$ and known or novel classes.

We design a loss function that constrains energy terms with positive training samples. For known-class samples, we expect lower known-class energy and higher novel-class energy, and vice versa. The loss function is given as:

$$L_{DE}^{eng} = \begin{cases} \lfloor E^k(\mathbf{x}) - E^l \rfloor + \lfloor E^h - E^u(\mathbf{x}) \rfloor, \mathbf{x} \text{ is known} \\ \lfloor E^h - E^k(\mathbf{x}) \rfloor + \lfloor E^u(\mathbf{x}) - E^l \rfloor, \mathbf{x} \text{ is novel} \end{cases}, \quad (12)$$

where E^l, E^h are parameters; $\lfloor \cdot \rfloor$ denotes $max(\cdot, 0)$.

Third, we design classification-based enhancement. In general, the training of classification tends to balance all categories with a weak ability to model known and novel classes. Thus, we design an extra binary classification head as an auxiliary, i.e., $f^b(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}$. During, training, the background samples are not included in optimizing f_b.

We use binary cross-entropy as the training loss:

$$L_{DE}^{bc} = \begin{cases} -\log(f^b(\mathbf{x})), \mathbf{x} \text{ is known} \\ -\log(1-f^b(\mathbf{x})), \mathbf{x} \text{ is novel} \end{cases}. \quad (13)$$

Overall, the loss function of OpenSLD is formulated as:

$$\begin{aligned} L = &\lambda_{FRPN} L_{FRPN} + \lambda_{IoU} L_{IoU} + \lambda_{loc} L_{loc} + \lambda_{cls} L_{cls} \\ &+ \lambda_{DE}(L_{DE}^{sim\_cls} + L_{DE}^{sim\_dist} + L_{DE}^{eng} + L_{DE}^{bc}) \end{aligned}, \quad (14)$$

where λ is the balance item, and L_{cls}, L_{loc} are classification and localization losses [41].

3.3 Novel-Class Subdivision with DetClust

OpenSLD treats all novel objects as a single "unknown" category without subdividing objects into different novel classes. To address this issue, we introduce a clustering method, called DetClust, which measures cosine similarity between different samples driven by BSDet's representation. Hence, the sample features have an inherent relation and discrimination based on BSDet, which helps improve the accuracy of clustering.

The DetClust algorithm can be divided into two parts: 1) initialization of the cluster centers and 2) optimization of the final clusters. Because of the varying number of novel categories, an adaptive method is designed to determine the number of clusters and select the initial cluster centers.

We introduce the cosine density to measure the distance of a sample from the cluster center. Figure 5 illustrates the sample distribution with three clusters, and darker colors represent higher cosine densities. The calculation of cosine density is based on RoI features $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$ from BSDet. As shown

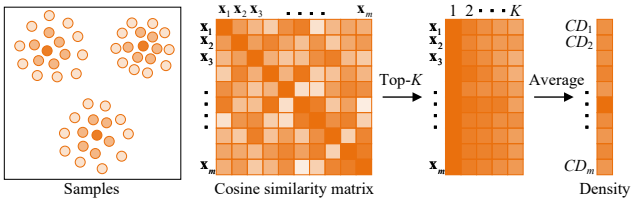

Fig. 5 The computation of cosine density.

in Figure 5, we calculate the pair-wise cosine similarities of the samples to obtain the cosine similarity matrix CS. Then, we use the average of top-K values as the density CD:

$$CS_{i,j} = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{||\mathbf{x}_i|| \cdot ||\mathbf{x}_j||}, \quad CD_i = \frac{\sum_{j=1}^{K} CS_{i,j}^{\text{top-}K}}{K}. \tag{15}$$

Next, to select the first initial cluster center from the samples, we define the cluster center score $score\_c$ to represent the confidence that a sample belongs to the initial cluster center. We use cosine density to initialize $score\_c$, i.e., $score\_c^1 = CD$, and leverage the samples with the highest $score\_c^1$ as the first cluster centers $\mathbf{c}_1$:

$$i_c^1 = \underset{i \in \{1,2,\ldots,m\}}{\text{argmax}} \; score\_c_i^1, \quad \mathbf{c}_1 = \mathbf{x}_{i_c^1}. \tag{16}$$

Then, the remaining initial cluster centers are selected iteratively. Note that the probability of a sample belonging to a new cluster is negatively correlated with the cosine similarity between the sample and the existing cluster centers. Therefore, we update $score\_c$ with previously obtained centers:

$$score\_c_i^{iter} = score\_c_i^{iter-1} \cdot \left(1 - \max(0, CS_{i,i_c^{iter-1}})^2\right), \tag{17}$$

where $iter > 1$ is the iteration step. Based on the above operation, the score of samples that are close to existing centers is suppressed. Hence, a new cluster center can be given as:

$$i_c^{iter} = \underset{i \in \{1,2,\ldots,m\}}{\text{argmax}} \; score\_c_i^{iter}, \quad \mathbf{c}_{iter} = \mathbf{x}_{i_c^{iter}}. \tag{18}$$

Finally, the iteration stops, if $\max_i score\_c_i^{iter} < T^c$, where T^c is a pre-defined threshold.

Furthermore, an optimization process is conducted on the initial cluster centers based on cosine similarity and the K-means algorithm [12] to obtain the final cluster centers. Then, the cluster results l^u are produced. The workflow of the DetClust algorithm is shown in the Appendix.

3.4 Few-Shot Incremental Learning

Online-discovered categories typically contain only a limited number of samples (i.e., a few-shot scenario), which could induce catastrophic forgetting during incremental learning. We employ the iCaRL method [13] to learn discovered few-shot samples with the following strategies to preserve the known knowledge.

Gradient Rescaling. Figure 6 illustrates the flow of BSDet, where black and red arrows represent the forward propagation and gradient backpropagation, respectively. Each module has a distinct role: 1) the backbone network is responsible for extracting features from the input image, 2) the RPN predicts region proposals, and 3) the detection head handles classification and localization predictions. During pre-training on known classes, a large amount of data is included. Thus, the features extracted by the backbone involve general visual and semantic information with class-agnostic properties. However, for incremental learning on novel-class samples, the small number of samples could lead to overfitting, resulting in catastrophic forgetting of known-class knowledge. Unlike the backbone, the RPN and detection head are directly related to the detection results and have fewer parameters. Therefore, the RPN and detection head should be primarily used to adapt to novel classes, while the backbone network should be slightly updated to maintain the robustness of feature extraction.

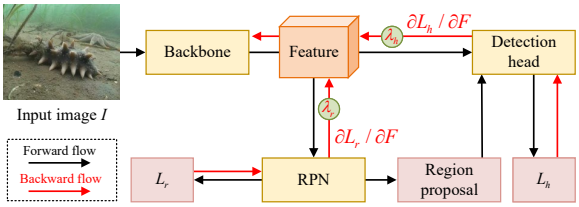

Fig. 6 Gradient rescaling for BSDet during the incremental learning.

We denote the parameters of backbone, RPN, and detection head as $\theta_b, \theta_r, \theta_h$ and loss functions for RPN and detection head as L_r, L_h. During training, it is obvious that θ_r and θ_h are only impacted by L_r and L_h, respectively. However, the gradient of the backbone is affected by both loss functions that can be given as:

$$\nabla_{\theta^b} = \frac{\partial L_r}{\partial \theta_b} + \frac{\partial L_h}{\partial \theta_b} = \frac{\partial L_r}{\partial F}\frac{\partial F}{\partial \theta_b} + \frac{\partial L_h}{\partial F}\frac{\partial F}{\partial \theta_b}. \tag{19}$$

where F is the backbone feature. With the gradient, parameters can be updated with a learning rate γ:

$$\theta_b = \theta_b - \gamma \nabla_{\theta_b} = \theta_b - \gamma \left(\frac{\partial L_r}{\partial F}\frac{\partial F}{\partial \theta_b} + \frac{\partial L_h}{\partial F}\frac{\partial F}{\partial \theta_b}\right). \tag{20}$$

To better control the influence of L_r and L_h on θ_b, we introduces gradient rescaling factors λ_r, λ_r during the back-propagation to the backbone network as follows,

$$\begin{aligned} \theta_b &= \theta_b - \gamma \left(\lambda_r \frac{\partial L_r}{\partial F}\frac{\partial F}{\partial \theta_b} + \lambda_h \frac{\partial L_h}{\partial F}\frac{\partial F}{\partial \theta_b}\right) \\ &= \theta_b - \gamma \left(\lambda_r \frac{\partial L_r}{\partial F} + \lambda_h \frac{\partial L_h}{\partial F}\right)\frac{\partial F}{\partial \theta_b} \end{aligned}. \tag{21}$$

That is, the gradients from L_r and L_h are scaled by factors at F, as shown in Figure 6.

Knowledge Distillation. In incremental learning methods, knowledge distillation on features or classification prediction is usually used to maintain the performance on base tasks [34, 45]. However, these methods usually require the inference of the base model during optimization, increasing

training costs. To tackle this problem, we devise knowledge distillation specifically for the representative vectors of BSDet to preserve the representations of known classes. In detail, the representative vectors trained on the known classes are used as the teacher vectors $\mathbf{tr} = \{\mathbf{tr}_1, \mathbf{tr}_2, \ldots, \mathbf{tr}_{N_k}\}$, while those in incremental learning are student vectors $\mathbf{r} = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{N_k}, \mathbf{r}_{N_k+1}, \mathbf{r}_{N_k+2}, \ldots, \mathbf{r}_{N_n}\}$, where N_k, N_n are the number of known and novel classes. Only $\mathbf{r}$ can be updated during incremental learning, while $\mathbf{tr}$ remains unchanged.

For distillation constraints, updated known-class vectors should be well-preserved, and loss functions can be designed using cosine similarity:

$$L_{rep} = \sum_{i=1}^{N_k} \left(1 - \frac{\mathbf{r}_i \cdot \mathbf{tr}_i}{||\mathbf{r}_i|| \cdot ||\mathbf{tr}_i||}\right). \tag{22}$$

Besides, the relative relationships among known-class vectors include the distribution information of known categories, so we constrain the similarity between pair-wise vectors:

$$\begin{aligned} TS_{i,j} &= \frac{\mathbf{tr}_i \cdot \mathbf{tr}_j}{||\mathbf{tr}_i|| \cdot ||\mathbf{tr}_j||}, SS_{i,j} = \frac{\mathbf{r}_i \cdot \mathbf{r}_j}{||\mathbf{r}_i|| \cdot ||\mathbf{r}_j||} \\ L_{rep\_mat} &= \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} (TS_{i,j} - SS_{i,j})^2 \end{aligned}. \tag{23}$$

Thereafter, L_{rep}, L_{rep_mat} are added to the optimization objective of BSDet during incremental learning.

4 Experiments

We first evaluate each module separately. Nevertheless, our method is not a simple combination of isolated components, and thus module-wise metrics do not adequately capture the performance of the full system. We consequently assess the entire pipeline via real-world underwater experiments.

4.1 OpenSLD

The PASCAL VOC [46] and MS COCO [47] datasets are used as benchmarks. PASCAL VOC has 20 categories, while MS COCO includes 80 classes encompassing 20 VOC categories. The training and validation sets of PASCAL VOC are used to train the OpenSLD model, and then the MS COCO validation set and the PASCAL VOC test set are used for testing. The 20 categories from VOC are treated as known classes, while the remaining 60 categories in MS COCO are novel classes. This experiment protocol is fully consistent with *Task 1* of OW-DETR [11] and OWOD [19] for fair comparison. Note that we do not report *Tasks 2∼4* like OW-DETR since their settings are not aligned with our O^2Exp pipeline. That is, *Tasks 2∼4* primarily evaluate standard (non-few-shot) incremental learning with substantial data per novel class, which conflicts with our intended scenario and few-shot setting, i.e., novel classes are learned with limited samples.

Table 1 Open-set detection results on COCO and VOC datasets

Method	WI ↓	A-OSE ↓	U-Recall ↑	mAP ↑
Faster R-CNN	0.0699	13396	–	56.2%
ORE\EBUI	0.0561	12064	4.9%	56.4%
OW-DETR	0.0571	10240	7.5%	59.2%
GroundingDINO	0.0882	14267	–	62.4%
DINO-X	0.0977	15363	–	**63.7%**
OpenSLD (ours)	**0.05319**	**8625**	**31.2%**	56.1%

During the training phase, the images are resized while maintaining their aspect ratio, with the width or height $\leqslant 1333$ or 800, respectively. We use an SGD optimizer with a momentum of 0.9 and a weight decay of 0.0001. The initial learning rate is set to $0.005 \times bs$, where $bs = 8$ is the batch size. The training process for OpenSLD spans 14 epochs, with the learning rate multiplied by 0.1 at the 10th and 13th epochs. More hyperparameters of OpenSLD are listed in the Appendix.

Closed-set object detection typically uses mean Average Precision (mAP) [46] to measure the detection accuracy of a model on known classes. However, in open-set object detection, it is also important to evaluate model performance in detecting novel objects. Specifically, during evaluation, the labels for 60 novel classes are integrated into a single "unknown" category. First, the unknown recall (U-Recall) [11] is used to measure the recall ability for novel objects. Additionally, it is important to consider the issue of confusion between known and novel objects during inference, where the novel objects could be mistakenly detected as known objects. For this issue, we use Absolute Open-Set Error (A-OSE) and Wilderness Impact (WI) [8] to measure the ability to distinguish between known and novel objects.

We compare our method with Faster R-CNN [41], ORE [9], OW-DETR [11], GroundingDINO [48], and DINO-X [49] in Table 1. ORE\EBUI is the ORE method with the removal of its EBUI module, so as to avoid the need for novel-class annotations in the test set. GroundingDINO and DINO-X are recent open-set detectors with text prompts, and we use known- and novel-class names as prompts for this testing. It is noted that the open-vocabulary methods require explicit text prompts to recognize objects. In the context of autonomous underwater exploration, where novel objects are often "unnamed" or "unseen" by human operators, prompt-based models face inherent limitations. In addition, our U-Recall is significantly higher than that of other existing

Table 2 Ablation study of FRPN on COCO and VOC datasets

FRPN		Metrics		
Classification	Centerness	WI ↓	A-OSE ↓	U-Recall ↑
	✓	0.05572	9020	25.7%
✓		0.05762	9751	24.6%
✓	✓	**0.05319**	**8625**	**31.2%**

Table 3 Ablation study of DEM on COCO and VOC datasets

DEM			Metrics		
L_{DEM}^{sim}	L_{DEM}^{eng}	L_{DEM}^{bc}	WI ↓	A-OSE ↓	U-Recall ↑
			0.06375	8977	23.8%
✓			0.05807	9508	28.4%
✓	✓		0.05362	9799	30.7%
✓	✓	✓	**0.05319**	**8625**	**31.2%**

methods, ensuring that the AUV can autonomously flag potential novelties for subsequent clustering and learning, which is a more practical paradigm for in-the-wild deployment. Also, OpenSLD outperforms existing methods in WI and A-OSE, showing its capability to distinguish novel objects from known categories, and our OpenSLD maintains on-par detection accuracy in mAP for known classes when compared to the traditional close-set detector Faster R-CNN.

The main difference between FRPN and RPN lies in the use of fused semantics and localization cues for foreground prediction. Thereby, we evaluate the performance based on either classification, centerness, or fused method. As shown in Table 2, the fusion method is more beneficial for detecting novel classes. Note that the U-Recall based on centerness is higher than that based on classification. This is because the centerness involves localization information with class-agnostic property, providing better generalization to novel classes. Moreover, PLS becomes more accurate as the performance of FRPN increases.

Table 3 presents the ablation results for DEM. It can be seen that when DEM is neglected, the performance is poor on the WI, A-OSE, and U-Recall. By incorporating the similarity-based enhancement, both WI and U-Recall show significant improvements. This indicates that the similarity-based design effectively enhances the ability to distinguish known or novel objects and the recall rate for novel categories. Although A-OSE increases after similarity learning, the decreased WI indicates that our similarity-based design is helpful. Moreover, when energy-based enhancement is added, it can be observed that both WI and U-Recall are further improved, validating the effectiveness of the proposed energy-based method. Referring to Figure 7, we calculate the Helmholtz

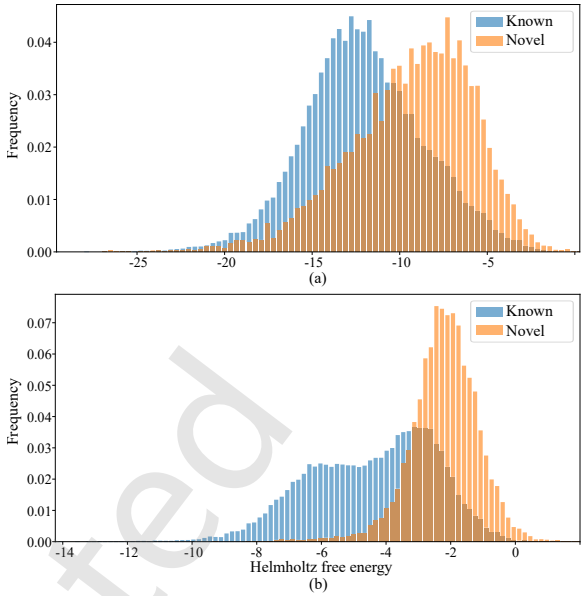

Fig. 7 The distribution of Helmholtz free energy. (a) w/o energy-based enhancement; (b) w/ energy-based enhancement. Our method leads to discriminative energy distributions between known and novel classes.

free energy of all samples in the test set. After adding the energy constraint, the energy distribution of both known and novel objects becomes more diverse, indicating that the method can increase the data discrimination from an energy perspective. Finally, classification-based enhancement enhances all metrics. Specifically, benefiting from the learning of foreground probability prediction, A-OSE is significantly improved. Please refer to the Appendix for qualitative results.

4.2 DetClust

We set $K = 20, T^c = 0.7$ and use the PASCAL VOC dataset for validation. To match the usage in the O²Exp framework, we conduct the validation of DetClust based on BSDet. Specifically, PASCAL VOC contains labels for 20 categories, and we designate "bird, bus, cow, motorbike, sofa" as novel classes, while the remaining 15 categories are known classes. First, we train BSDet using the known-class data and then extract features from novel-class images. Next, the annotation of the novel classes is used to perform RoI pooling on features, obtaining feature vectors of the novel-class objects. Finally, 200 sample features from each class are randomly selected, which are clustered using DetClust for evaluation. The evaluation process of DetClust is illustrated in the Appendix.

Since the cluster and ground-truth indices could be inconsistent, we use the Hungarian matching algorithm [50] to match the clustering results with the ground truth. Purity,

Table 4 Comparison of clustering methods on VOC dataset

Method	Cluster number	Purity ↑	RI ↑	F-score ↑
K-means	Manual	0.925	0.943	0.858
HC	Manual	0.905	0.930	0.825
SC	Manual	0.663	0.819	0.575
AP	60 (×)	–	–	–
DBSCAN	5 (✓)	0.767	0.870	0.696
DetClust (ours)	5 (✓)	**0.947**	**0.960**	**0.899**

HC: Hierarchical clustering; SC: Spectral clustering;
AP: Affinity propagation; ✓: correct estimation.

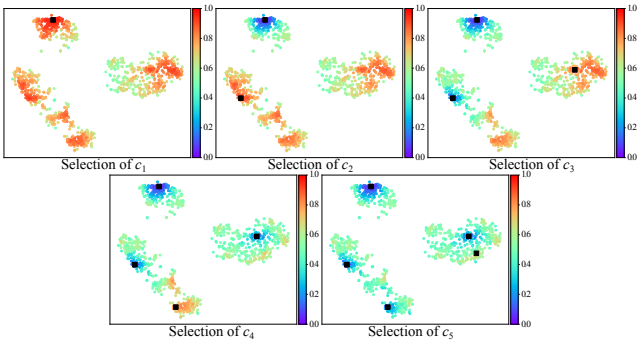

Fig. 8 Distribution of cluster center score across iterations. The black points are selected cluster centers, and the color bar indicates the score.

Rand Index (RI), and F-score are employed as the metrics. Also, t-SNE is leveraged to perform dimensionality reduction on sample features for visualization.

We compare the DetClust with K-means [12], hierarchical clustering[25], spectral clustering[51], affinity propagation[52], and DBSCAN [27], as shown in Table 4. K-means, hierarchical clustering, and spectral clustering require a pre-defined cluster number, whereas affinity propagation and DBSCAN are adaptive in cluster number. Experimental results show that DetClust can correctly estimate the number of clusters and achieve the best metrics, demonstrating the superiority of DetClust. Notably, when the number of clusters is manually given, K-means significantly outperforms other methods except for DetClust, indicating that the K-means algorithm is more suitable for the features extracted by BSDet. This also validates the effectiveness of using a K-means-inspired fine-tuning approach in DetClust. Furthermore, compared to K-means, DetClust can estimate initial cluster centers and measure similarity between samples based on cosine similarity, leading to better results. Among the methods that adaptively estimate the number of clusters, affinity propagation fails to predict the correct cluster number, while DBSCAN produces poor clustering results.

Figure 8 shows the DetClust estimation of the initial cluster centers, where the color indicates their cluster center scores and the black squares represent the initially selected centers. In the first iteration, the cluster center scores are from cosine density. As shown, the cosine density effectively indicates the proximity of samples to the cluster center, demonstrating the effectiveness of our design. Additionally, in each iteration, the changes in the cluster center scores show that the selected centers successfully suppress the scores of surrounding samples. This effectively prevents the issue of producing multiple centers for the same class. Finally, DetClust successfully selects 5 cluster centers, and the scores of all remaining samples are below the threshold.

After selecting the initial cluster centers, DetClust optimizes them based on the K-means algorithm. We refer readers to the appendix for the visualization of K-means-based optimization.

4.3 Few-Shot Incremental Learning

We use the PASCAL VOC dataset to validate the proposed incremental learning method, which is the standardized benchmark for our incremental learning experiments to ensure a direct and fair comparison with existing state-of-the-art few-shot incremental learning methods. While the VOC benchmark is relatively easy, they serve primarily for initial methodological validation, whereas the ultimate verification of our framework lies in the subsequent real-world underwater experiments. Classes of “bird, bus, cow, motorbike, sofa” are set as the novel classes for incremental learning, while the remaining 15 classes are considered as known classes. Unlike offline incremental methods that rely on a large number of samples [11], O2Exp evolves through autonomously discovered few-shot samples (from 1 to 10 shots), making it a more rigorous but practical task for AUVs. That is, we first train BSDet with the known-class data, and incrementally update BSDet with few-shot known and novel samples. After training, the model is tested on the VOC test set, evaluating the mAP on known and novel classes.

Referring to Table 5, we use the average of 1- to 10-shot mAP as the final result (Please refer to the Appendix for the full few-shot results.), where “KD” represents the use of distillation loss function. This ablation study follows an additive strategy. We first validated the effectiveness of the distillation constraint (comparing Rows 1 and 2), which serves as a foundational component for representation stability. Building upon this, we systematically tuned the gradient rescaling factors (λ_r, λ_h) to further suppress catastrophic forgetting while allowing for efficient adaptation to novel classes. We employed the additive ablation strategy as the two

Table 5 Results of few-shot incremental learning on VOC dataset. The result is the average mAP of 1- to 10-shot experiments. The first line shows the setting of iCaRL [13].

KD	λ_r	λ_h	Known-class (%)	Novel-class (%)
×	1.0	1.0	58.0	47.1
✓	1.0	1.0	59.2	48.3
✓	0.5	1.0	59.3	49.8
✓	0.1	1.0	59.3	49.9
✓	0.01	1.0	59.1	49.9
✓	0.001	1.0	59.3	49.4
✓	0	1.0	60.6	49.7
✓	0	0.5	66.4	51.3
✓	0	0.1	76.1	55.3
✓	0	0.01	79.8	**57.6**
✓	0	0.001	**80.5**	56.0
✓	0	0	**80.5**	55.4

modules target independent aspects of the network. That is, "KD" acts on the semantic space while λ acts on the feature space. Thus, their contributions are orthogonal and additive rather than non-linearly coupled.

Before incremental learning, the BSDet achieves a baseline of 80.6% on the known classes. In the first row, our baseline iCaRL [13] does not use the proposed strategies, resulting in the forgetting of the known-class knowledge and a decrease in known-class mAP. After adding the knowledge distillation loss constraint (the 2nd row), the mAP on the known classes increases, validating the effectiveness of the proposed knowledge distillation loss. Additionally, due to the improved ability to preserve known-class knowledge, the detector can better distinguish novel and known classes, resulting in an increased mAP on novel classes.

Rows 3-7 show the results under different values of λ_r, which is the rescaling factor for the gradients from the RPN propagated back to the backbone network. It can be observed that as λ_r decreases, there is a slight improvement in the mAP for both known and novel classes. Rows 8-12 show the results on both known and novel classes under different values of λ_h, which represents the rescaling factor for the gradients from the detection head propagated back to the backbone network. It can be seen that as λ_h decreases, the mAP for the known class improves significantly. When λ_h is extremely small or even zero, the mAP for known classes remains almost the same as the baseline. Note that when λ_h is too small, the parameters of the backbone network can hardly be updated. Despite the effective preservation of known-class representation, it hinders the learning efficiency of novel-class samples. The detection results with different values of λ_r and λ_h validate the effectiveness of the proposed gradient

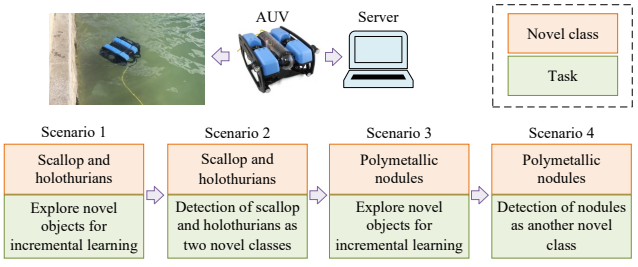

Fig. 9 Illustration of real-world tasks.

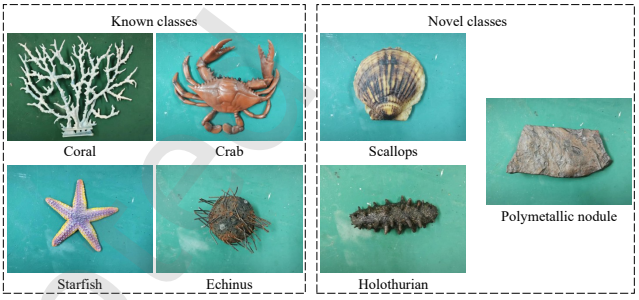

Fig. 10 Object categories in real-world experiments.

rescaling. This demonstrates that reducing the gradients back-propagated from the detection head and RPN to the backbone network significantly enhances the preservation of feature representations for known classes. Finally, we set $\lambda_r = 0$ and $\lambda_h = 0.01$.

4.4 Underwater Object Exploration

We apply our O²Exp framework to object exploration in a wild unstructured environment with an AUV platform. The system design and task flow are shown in Figure 9. The BlueROV2 AUV is employed for real-time data collection and transmission; the server with an Intel Core i7-9750H and an NVIDIA GeForce RTX 2080 is responsible for performing the O²Exp framework. Before real-world experiments, the O²Exp is pre-trained using the VOC and our collected underwater data. Our dataset includes seven categories: echinus, starfish, coral, crab, holothurian, scallop, and polymetallic nodule, where the first four are set as known classes and the latter three are set as novel categories (see Figure 10). During training, only known-class data is used to train detectors. See, we Figure 9, we design four experimental scenarios to verify that the O²Exp can continuously explore novel classes and perform incremental learning.

During online exploration, object detection and model update run in parallel. The BSDet and OpenSLD can run at 25FPS for online object detection. DetClust and incremental learning take approximately 300 seconds, so we set the update period $T^p = 300$. During this time, the detectors collect

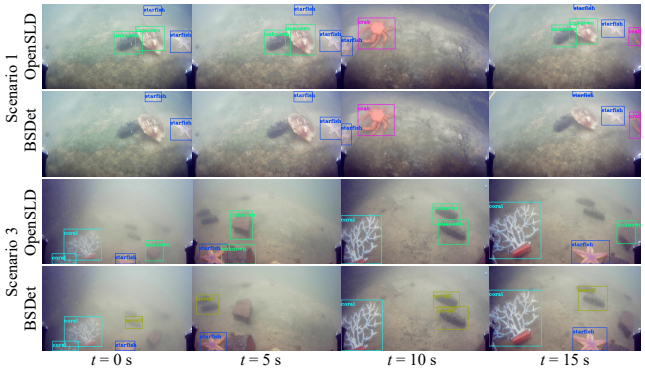

Fig. 11 Detection results in Scenarios 1 and 3.

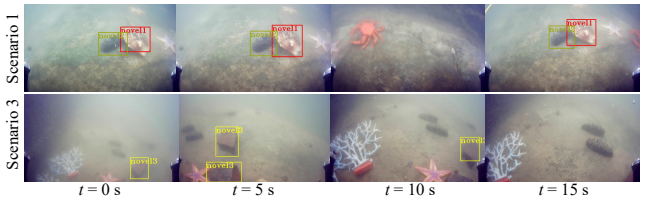

Fig. 12 Pseudo-labels produced by OpenSLD and DetClust in Scenarios 1 and 3.

data and retain samples with high confidence. Subsequently, DetClust generates labels for incremental learning to optimize and update model parameters. While incremental learning is ongoing, the detectors can synchronously perform the detection task.

The O^{2}Exp includes two detectors: an open-set OpenSLD and a closed-set BSDet. During the underwater exploration by AUV, only BSDet is involved in incremental learning, while all parameters in OpenSLD remain unchanged. Therefore, in all scenarios, the detection results of OpenSLD only include five classes: echinus, starfish, coral, crab, and novel class. On the other hand, BSDet can detect echinus, starfish, coral, crab, and multiple novel classes that have been incrementally learned.

In Scenario 1, the detection results of OpenSLD and BSDet are shown in Figure 11. It can be seen that OpenSLD is able to detect holothurians and scallops as novel-class objects. However, since BSDet has not yet learned these categories, it cannot detect either of them.

Then, the novel-class features are clustered using DetClust. Referring to Figure 12, holothurians and scallops are annotated as novel classes based on the DetClust results. By combining the results from Figure 11 and 12, the annotations for known and novel classes in Scenario 1 are obtained. These labeled images can be used as training samples for incremental learning. The detection results in Scenario 1 validate the effectiveness of O^{2}Exp in the detection and subdivision of novel classes.

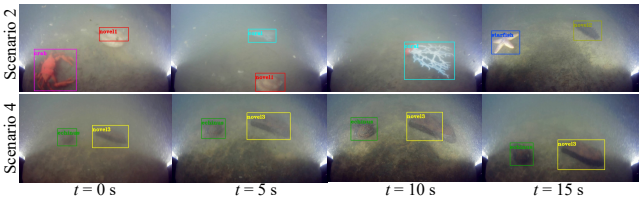

Fig. 13 Detection results of BSDet in Scenarios 2 and 4 after few-shot incremental learning.

After incremental learning, O^{2}Exp can perform object exploration in Scenario 2, as shown in Figure 13. As BSDet has already learned samples of holothurian and scallop in Scenario 1, it successfully detects them as novel classes in Scenario 2. Additionally, the known classes, e.g., crab and coral, can still be detected after incremental learning. It is noteworthy that corals did not appear in Scenario 1, so coral samples are absent in incremental learning. Nevertheless, BSDet can maintain the ability to detect corals after incremental learning. The results in Scenario 2 validate the effectiveness of the incremental learning method, demonstrating that BSDet can preserve the knowledge of known classes while learning novel knowledge in the meantime.

We perform experiments in Scenarios 3 and 4 with polymetallic nodules as the novel class, as shown in Figure 11–13. In Scenario 3, OpenSLD detects nodules and holothurians as novel classes, while holothurians are detected as known objects by BSDet. As a result, only nodules are selected as novel-class samples. After incremental learning, BSDet can detect nodules as another novel category.

The results from Scenarios 1 to 4 demonstrate O^{2}Exp can continuously explore novel-class objects and learn novel-class knowledge. This enables the AUV to accomplish autonomous visual perception in online object exploration tasks. Please refer to our supplementary video for a dynamic demonstration.

5 Conclusion and Future Works

This paper focuses on online object exploration for underwater environments and breaks this task into open-set detection, fine-grained novel-class subdivision, and incremental learning. We also propose the O^{2}Exp framework for this challenge. First, an OpenSLD method is proposed for detecting novel-class objects with semantics and localization cues. Second, a DetClust method is proposed to enable the adaptive determination of cluster numbers and the subdivision of novel-class objects. Third, gradient rescaling and knowledge distillation strategies are designed, allowing the detector to preserve base-class knowledge during incremental learning. Extensive experiments on public datasets and real-world environments demonstrate the effectiveness of the proposed framework,

showing that the AUV can effectively perform the online object exploration task in unstructured underwater scenarios.

In the future, we will investigate more real-world applications with the O^{2}Exp framework. Besides, we plan to improve the O^{2}Exp framework using an open-vocabulary paradigm.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant 62233001, Grant 62403012, Grant U24A20282, Grant 62473236) and Zhongguancun Academy (Grant C20250503).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

Reference

[1] F. Han, J. Yao, H. Zhu, C. Wang *et al.*, Underwater image processing and object detection based on deep CNN method, *Journal of Sensors*, 2020.

[2] M. Zhang, S. Xu, W. Song, Q. He, and Q. Wei, Lightweight underwater object detection based on YOLOv4 and multi-scale attentional feature fusion, *Remote Sensing*, vol. 13, no. 22, p. 4706, 2021.

[3] S. Kong, M. Tian, C. Qiu, Z. Wu, and J. Yu, IWSCR: An intelligent water surface cleaner robot for collecting floating garbage, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 10, pp. 6358–6368, 2021.

[4] Y. Yu, Z. Cao, Z. Liu, W. Geng, J. Yu, and W. Zhang, A two-stream cnn with simultaneous detection and segmentation for robotic grasping, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 1167–1181, 2022.

[5] G. Ren, J. Liu, M. Wang, P. Guan, Z. Cao, and J. Yu, Few-shot object detection via dual-domain feature fusion and patch-level attention, *Tsinghua Science and Technology*, vol. 30, no. 3, pp. 1237–1250, 2025.

[6] Z. Gong, X. Fang, X. Chen, J. Cheng, Z. Xie, J. Liu, B. Chen, H. Yang, S. Kong, Y. Hao *et al.*, A soft manipulator for efficient delicate grasping in shallow water: Modeling, control, and real-world experiments, *The International Journal of Robotics Research*, vol. 40, no. 1, pp. 449–469, 2021.

[7] R. Liu, X. Fan, M. Zhu, M. Hou, and Z. Luo, Real-world underwater enhancement: Challenges, benchmarks, and solutions under natural light, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 12, pp. 4861–4875, 2020.

[8] A. Dhamija, M. Gunther, J. Ventura, and T. Boult, The overlooked elephant of object detection: Open set, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020.

[9] K. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, Towards open world object detection, in *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference*, 2021.

[10] Y. Lu, X. Chen, Z. Wu, M. Tan, and J. Yu, Binary similarity few-shot object detection with modeling of hard negative samples, *IEEE Transactions on Multimedia*, vol. 26, pp. 4805–4818, 2023.

[11] A. Gupta, S. Narayan, K. Joseph, S. Khan, F. S. Khan, and M. Shah, OW-DETR: Open-world detection transformer, in *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference*, 2022.

[12] A. Ahmad and L. Dey, A K-mean clustering algorithm for mixed numeric and categorical data, *Data & Knowledge Engineering*, vol. 63, no. 2, pp. 503–527, 2007.

[13] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, iCaRL: Incremental classifier and representation learning, in *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference*, 2017.

[14] A. Jesus, C. Zito, C. Tortorici, E. Roura, and G. De Masi, Underwater object classification and detection: first results and open challenges, *OCEANS*, 2022.

[15] X. Chen, Y. Lu, Z. Wu, J. Yu, and L. Wen, Reveal of domain effect: How visual restoration contributes to object detection in aquatic scenes, *arXiv:2003.01913*, 2020.

[16] X. Chen, J. Yu, S. Kong, Z. Wu, and L. Wen, Joint anchor-feature refinement for real-time accurate object detection in images and videos, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 2, pp. 594–607, 2021.

[17] B. Fan, W. Chen, Y. Cong, and J. Tian, Dual refinement underwater object detection network, in *Proceedings of the European Conference on Computer Vision*, 2020.

[18] X. Hua, X. Cui, X. Xu, S. Qiu, Y. Liang, X. Bao, and Z. Li, Underwater object detection algorithm based on feature enhancement and progressive dynamic aggregation strategy, *Pattern Recognition*, vol. 139, p. 109511, 2023.

[19] K. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, Towards open world object detection, in *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference*, 2021.

[20] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, End-to-end object detection with transformers, in *Proceedings of the European Conference on Computer Vision*, 2020.

[21] D. Kim, T.-Y. Lin, A. Angelova, I. S. Kweon, and W. Kuo, Learning open-world object proposals without learning to classify, *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5453–5460, 2022.

[22] Z. Wu, Y. Lu, X. Chen, Z. Wu, L. Kang, and J. Yu, UC-OWOD: Unknown-classified open world object detection, in *Proceedings of the European Conference on Computer Vision*, 2022.

[23] H.-S. Park and C.-H. Jun, A simple and fast algorithm for K-medoids clustering, *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[24] M. Van der Laan, K. Pollard, and J. Bryan, A new partitioning around medoids algorithm, *Journal of Statistical Computation and Simulation*, vol. 73, no. 8, pp. 575–584, 2003.

[25] F. Murtagh and P. Contreras, Algorithms for hierarchical clustering: an overview, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[26] ——, Methods of hierarchical clustering, *arXiv:1105.0121*, 2011.

[27] O. Kramer and H. Danielsiek, DBSCAN-based multi-objective niching to approximate equivalent pareto-subsets, in *Proceedings of the Annual Conference on Genetic and Evolutionary Computation*, 2010.

[28] K. M. Kumar and A. R. M. Reddy, A fast dbscan clustering algorithm by accelerating neighbor searching using groups method, *Pattern Recognition*, vol. 58, pp. 39–48, 2016.

[29] W. Wang, J. Yang, R. Muntz *et al.*, Sting: A statistical information grid approach to spatial data mining, in *Proceedings of the International Conference on Very Large Data Bases*, 1997.

[30] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, Structural deep clustering network, in *Proceedings of the Web Conference*, 2020.

[31] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, Deep clustering for unsupervised learning of visual features, in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 132–149.

[32] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, Few-shot class-incremental learning, in *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference*, 2020.

[33] P. Singh, P. Mazumder, P. Rai, and V. P. Namboodiri, Rectification-based knowledge retention for continual learning, in *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference*, 2021.

[34] Z. Li and D. Hoiem, Learning without forgetting, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[35] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, PODNet: Pooled outputs distillation for small-tasks incremental learning, in *Proceedings of the European Conference on Computer Vision*, 2020.

[36] P. Zhou, L. Mai, J. Zhang, N. Xu, Z. Wu, and L. S. Davis, M2KD: Multi-model and multi-level knowledge distillation for incremental learning, *arXiv:1904.01769*, 2019.

[37] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, Learning a unified classifier incrementally via rebalancing, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 831–839.

[38] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, Large scale incremental learning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 374–382.

[39] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, and J. v. d. Weijer, Semantic drift compensation for class-incremental learning, in *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference*, 2020.

[40] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, and J. Choi, Rainbow memory: Continual learning with a memory of diverse samples, in *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference*, 2021.

[41] S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in *Proceedings of the Advances in Neural Information Processing Systems*, 2015.

[42] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, Feature pyramid networks for object detection, in *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference*, 2017.

[43] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, A tutorial on energy-based learning, *Predicting Structured Data*, vol. 1, no. 0, 2006.

[44] G. E. Hinton and R. Zemel, Autoencoders, minimum description length and helmholtz free energy, in *Proceedings of the Advances in Neural Information Processing Systems*, 1993.

[45] L. Chen, C. Yu, and L. Chen, A new knowledge distillation for incremental object detection, in *Proceedings of the International Joint Conference on Neural Networks*, 2019.

[46] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, The pascal visual object classes (VOC) challenge, *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2010.

[47] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, Microsoft COCO: Common objects in context, in *Proceedings of the European Conference on Computer Vision*, 2014.

[48] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, Grounding dino: Marrying dino with grounded pre-training for open-set object detection, in *Proceedings of the European Conference on Computer Vision*, 2024.

[49] T. Ren, Y. Chen, Q. Jiang, Z. Zeng, Y. Xiong, W. Liu, Z. Ma, J. Shen, Y. Gao, X. Jiang *et al.*, DINO-X: A unified vision model for open-world object detection and understanding, *arXiv preprint arXiv:2411.14347*, 2024.

[50] H. W. Kuhn, The Hungarian method for the assignment problem, *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[51] U. Von Luxburg, A tutorial on spectral clustering, *Statistics and Computing*, vol. 17, pp. 395–416, 2007.

[52] K. Wang, J. Zhang, D. Li, X. Zhang, and T. Guo, Adaptive affinity propagation clustering, *arXiv:0805.1096*, 2008.

Author biography

Xingyu Chen received the B.E. degree in electrical engineering and automation from the College of Nuclear Technology and Automation Engineering, Chengdu University of Technology, Chengdu, China, in 2015, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences (IACAS), Beijing, China, in 2020.

From 2024 to 2025, he was a Post-Doctoral Research Fellow with the School of Advanced Manufacturing and Robotics, Peking University, Beijing. He is currently an Assistant Professor with Zhongguancun Academy. His research interest is embodied intelligence.

Yue Lu received the B.E. degree in electrical information science and technology from the College of Electronic Science and Engineering, Jilin University, Changchun, China, in 2018, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences (IACAS), Beijing, China, in 2023.

His research interests include computer vision, deep learning, and underwater robotics.

Shaoan Wang eceived the B.E. degree in mechanical engineering from the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China, in 2021. He is currently pursuing the Ph.D. degree in general mechanics and foundation of mechanics with the School of Advanced Manufacturing and Robotics, Peking University, Beijing, China. His current research interests include embodied AI and navigation.

Zhengxing Wu received the B.E. degree in logistics engineering from the School of Control Science and Engineering, Shandong University, Jinan, China, in 2008, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences (IACAS), Beijing, China, in 2015.

He is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, IACAS. His research interests include bioinspired robots and intelligent control systems.

Junzhi Yu received the B.E. degree in safety engineering and the M.E. degree in precision instruments and mechanology from the North University of China, Taiyuan, China, in 1998 and 2001, respectively, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2003.

From 2004 to 2006, he was a Post-Doctoral Research Fellow with the Center for Systems and Control, Peking University, Beijing. He was an Associate Professor with the Institute of Automation, Chinese Academy of Sciences, in 2006, where he was a Full Professor in 2012. In 2018, he joined the College of Engineering, Peking University, as a Tenured Full Professor. His current research interests include intelligent robots, motion control, and intelligent mechatronic systems.